

2016-2017

*Projet 'Développement d'une
application en robotique mobile'*

Anrouz Ayoub

Derraz Sofiane

Xiaoming Wang

Nantenaina Andriamiraho

MARECHAL Pierre

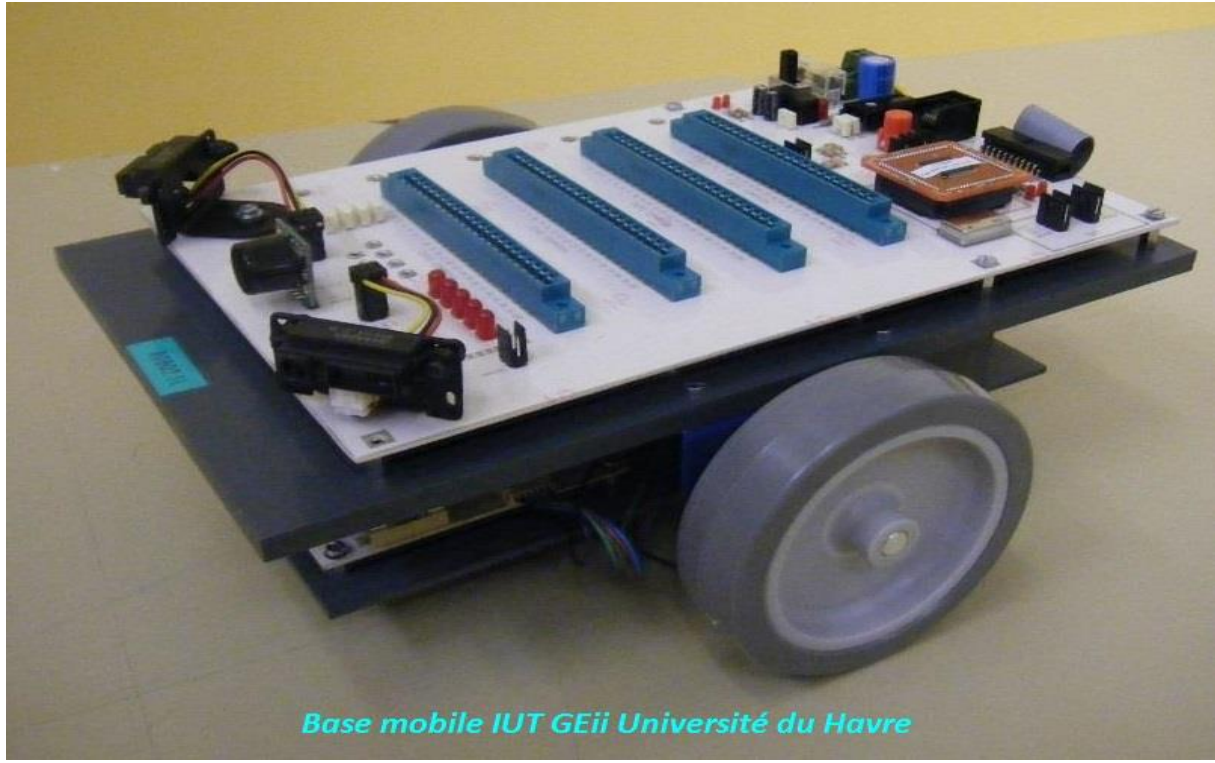
PELVILLAIN Hervé



Sommaire

Introduction.....	p.2
Présentation de la base mobile.....	p.3-5
1. Module chargeur de batterie :	
2. Module motorisation & commande des moteurs :	
3. Module microcontrôleur :	
4. Etude des capteurs :	
Schéma structurelle.....	p.6-8
-Carte1	
-Carte2	
Typon de la carte2.....	p.9
Programmation.....	p.10-11
Annexes.....	p.12

Introduction :



Les premiers robots mobiles sont apparus dans le monde industriel au début des années 1970. Aujourd'hui, ces derniers font partie de notre quotidien et pénètrent des secteurs d'activités de plus en plus diversifiés tels que l'exploration sous-marine, l'espace l'aéronautique (drones, dirigeables), les travaux publics, l'agriculture, le déminage ou l'intervention sur sites dangereux. Pour chacun de ces domaines d'application, le robot mobile a pour fonction d'aider l'homme lors de la réalisation de tâches répétitives et pénibles qui sont réalisées le plus souvent en milieux hostiles.

PRESENTATION de la BASE MOBILE

La base mobile est constituée de:

Une carte chargeur de batterie



Une batterie plomb 12V 1,2 AH



Deux moteurs à courant continu



Une carte mère



1. Module chargeur de batterie :

La carte chargeur de batterie est équipée d'un circuit de commutation automatique piloté par une alimentation secteur 230 Vac / 24 Vdc.

En mode CHARGE, base mobile sur cale, la batterie n'est plus sollicitée et est chargée à 1/10 de sa capacité par un processeur de charge spécialisé (UC3906)

- ➡ LED rouge = Mode CHARGE
- ➡ LED verte = Batterie en charge.

2. Module motorisation & commande des moteurs :

PWM : La modulation de largeur d'impulsions, est une technique couramment utilisée pour synthétiser des signaux continus à l'aide de circuits à fonctionnement tout ou rien, ou plus généralement à états discrets.

Le principe général est qu'en appliquant une succession d'états discrets pendant des durées bien choisies, on peut obtenir en moyenne sur une certaine durée n'importe quelle valeur intermédiaire.

Chaque motoréducteur EMG30 est commandé en puissance par un circuit dédié à cette fonction: LMD18200 (pont en H)

La variation de la vitesse de rotation est définie par le pourcentage de la commande PWM (Pulse Width Modulation – Modulation de la largeur d'impulsion) générée par le µcontrôleur de la carte mère (module 3)

- ➡ PWM = 0 % Moteur à l'arrêt
- ➡ PWM = 50 % Moteur en ½ vitesse
- ➡ PWM = 100 % Moteur à vitesse maximum

3. Module microcontrôleur :

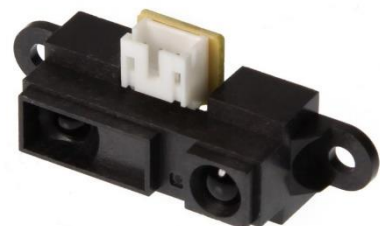
- ➡ Microcontrôleur dSPIC 30F6010A, bouton RESET et interface de programmation pour PICKIT3 ou ICD3
- ➡ Oscillateur 20MHZ
- ➡ La communication des données de commande avec le module 2 s'effectue via un câble en nappe gris de 20 conducteurs.

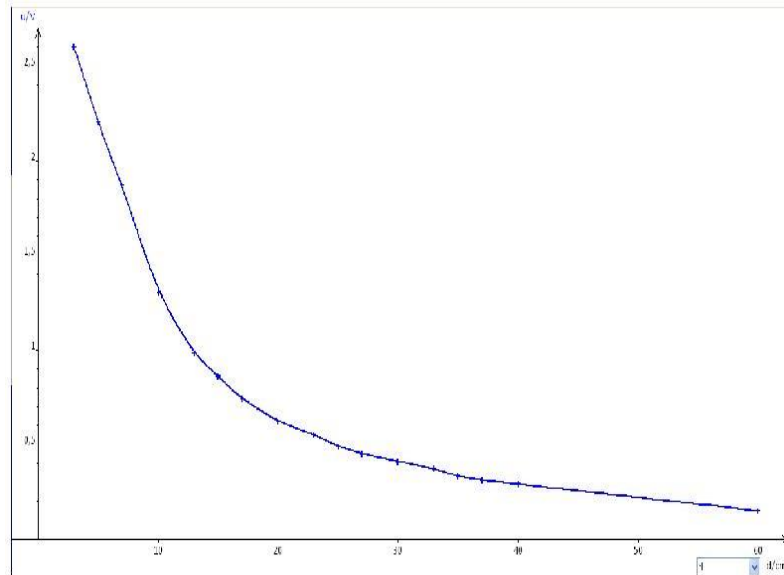
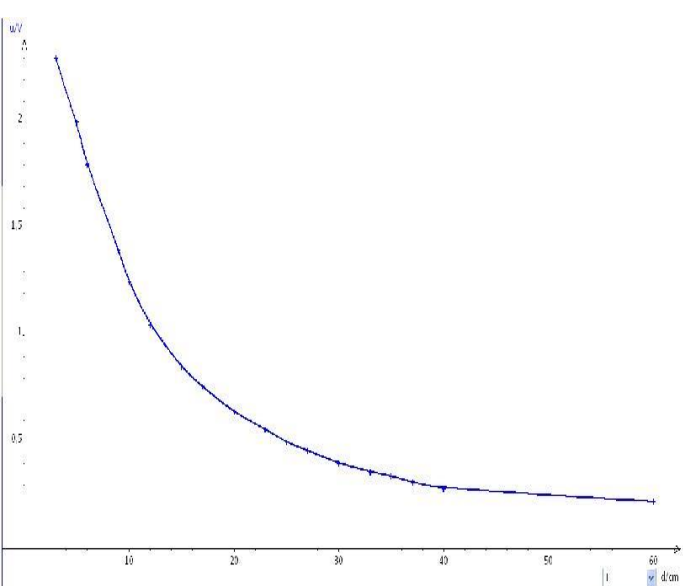
4. Etudes des capteurs :

➡ Deux télémètres infra rouges SHARP™ GP2Y

Un capteur infrarouge est un détecteur réagissant à un rayonnement infrarouge (IR).

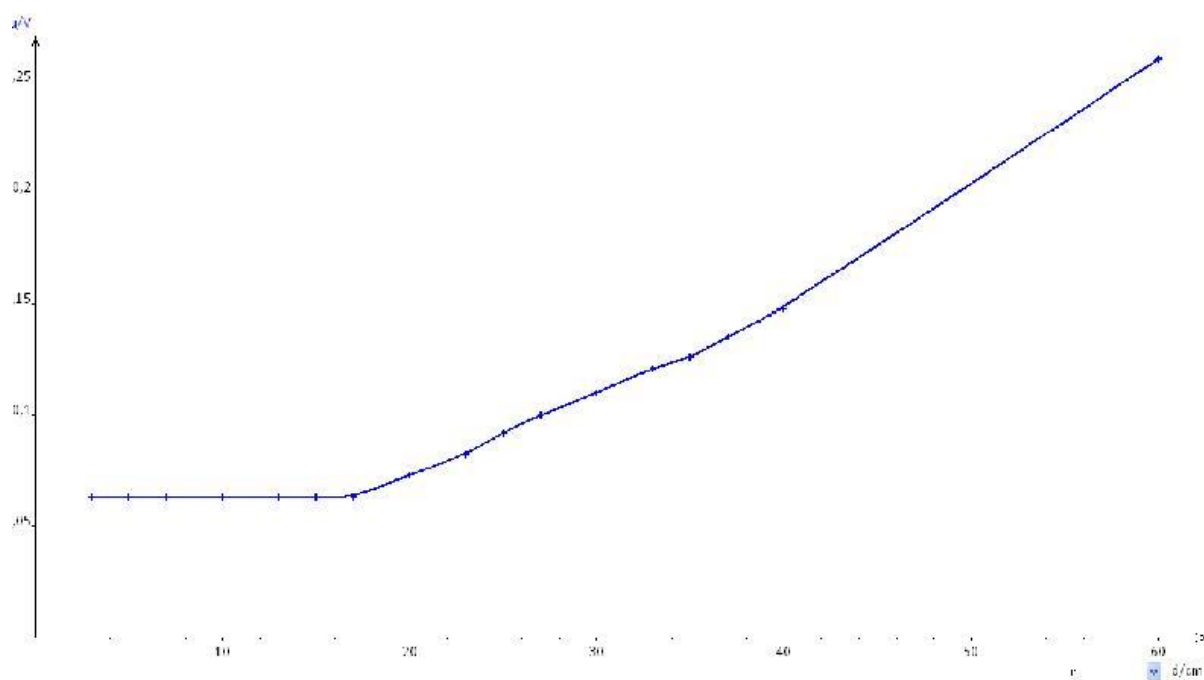
Le principe d'un capteur par réflexion est d'éclairer l'obstacle avec une LED infrarouge





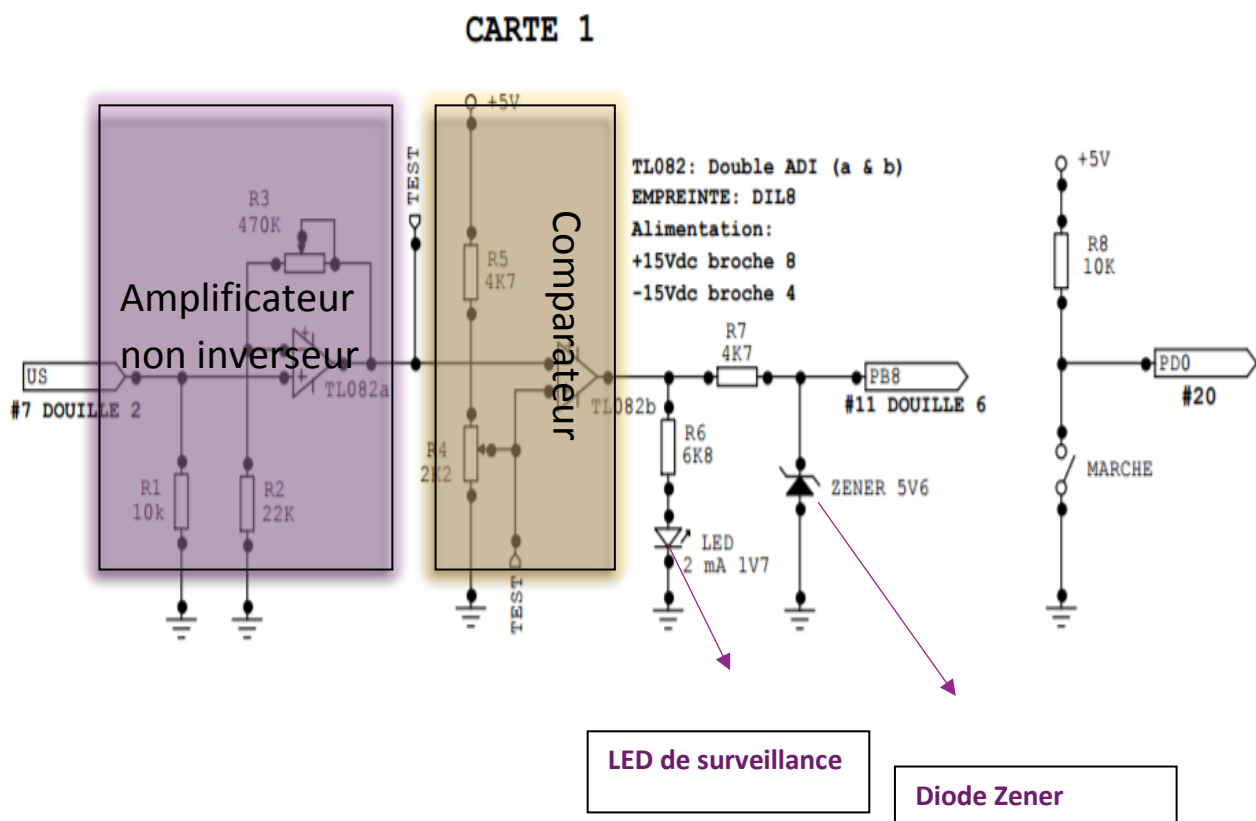
→ Un télémètre à ultrasons LV-MaxSonar™ EZ1

Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho.



CHEMA STRUCTURELLE

Carte1 : capteur Ultrasons



- **Amplificateur non inverseur :**

$$V_{out} = \frac{9.8mV}{2.54cm}$$

Pour une distance de 20cm → $V_{out}=78mV$.

La tension de sortie V_{test} est comprise entre 78mV ($R3=470K$) et 1.8V ($R3=0$)

$$78mV < V_{test} < 1.8V$$

La fonction du transfert :

$$V_{test} = \frac{R2+R3}{R2} * V_e$$

L'amplificateur non inverseur génère la tension venue par le capteur après l'avoir amplifiée et elle est envoyée sur l'entrée (-) d'un comparateur.

- **Comparateur :**

Le Comparateur compare deux tensions et renvoie à la sortie une tension soit $+V_{cc}$ soit $-V_{cc}$

A l'aide du potentiomètre R4 on fixe une tension limite qui correspond à la distance 25cm (environ 78mV) et on la compare à la tension renvoyée par le capteur (V_{test})

Si $V > V_+$ → Sortie $-V_{cc}$ le robot est loin de l'obstacle

Si $V < V_+$ → Sortie $+V_{cc}$ le robot est proche de l'obstacle

- **La LED :**

La Led sert à visualiser l'état de la sortie et de savoir si le courant qui passe est positif.

Elle est allumée quand la sortie du comparateur est $+V_{cc}$ (on est près de l'obstacle).

- **Diode Zener :**

La diode zener sert à stabiliser le courant.

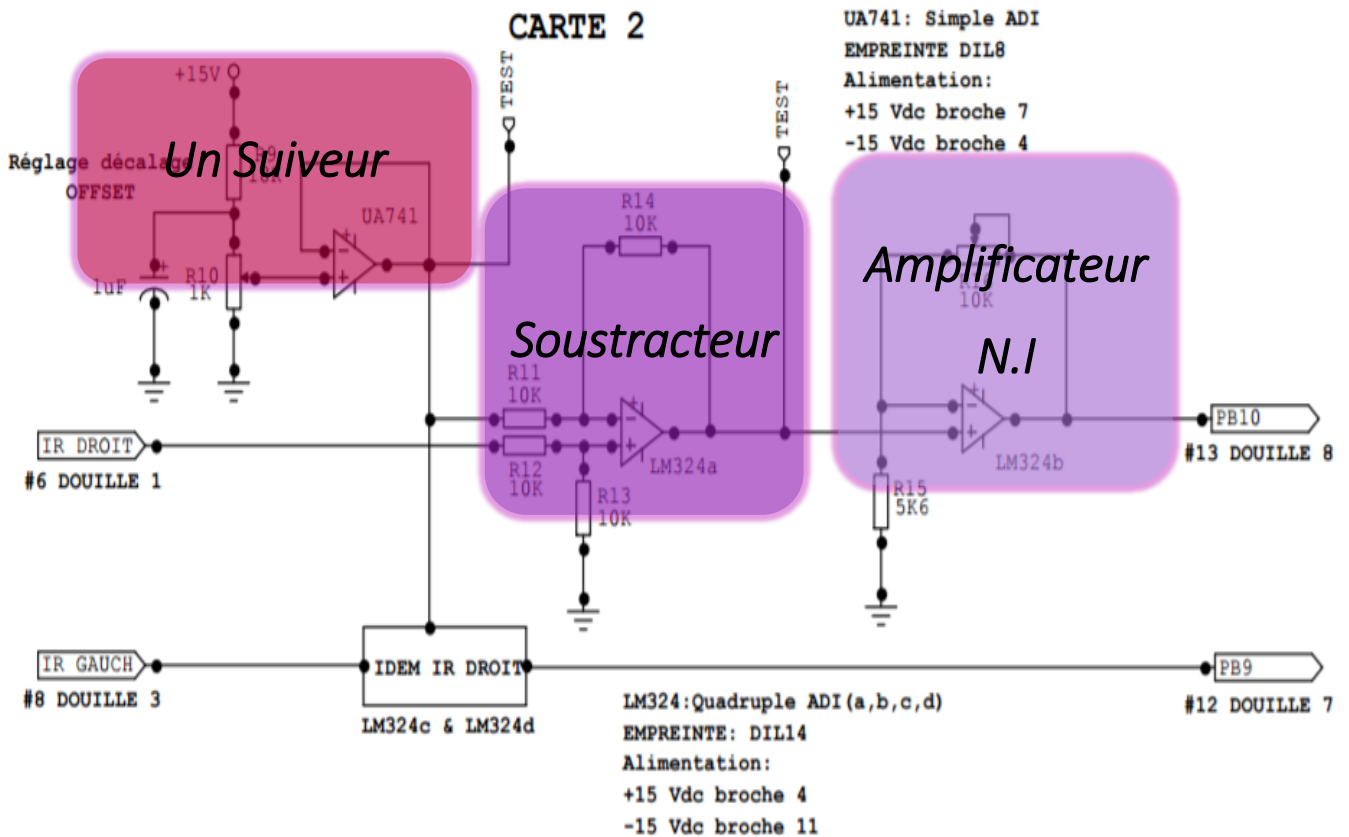
Si la tension sortie du comparateur est supérieure à 5.6V, la diode sera à l'état bloqué.

Si la tension sortie du comparateur est inférieure à 5.6V, la diode sera passante.

(La résistance R7 empêche la surtension dans la diode avec une petite valeur)

(La résistance R6 sert à protéger la Led)

Carte2 : capteur Infrarouge



- *Suiveur* :

Pour régler la tension de l'offset on utilise tout d'abord un suiveur qui recopie la tension de l'entrée sur la sortie.

Le potentiomètre sert à régler une tension qui sera soustraite après pour ça on modifie R10 afin d'obtenir une tension 22mV.

La capacité corrige le signal en entré avant de l'envoyer au suiveur.

- *Soustracteur :*

Le soustracteur sert à enlever la valeur de l'offset envoyée au début.

La tension du capteur droit est envoyée sur l'entré (+).

La tension de l'offset est envoyée sur l'entré (-).

$$V_+ > V_- \rightarrow V_{ir} > V_{offset} \rightarrow \text{proximité de l'obstacle}$$

$V_+ < V_- \rightarrow V_{ir} < V_{offset} \rightarrow$ loin de l'obstacle

Si la sortie du soustracteur est 0 cela veut dire que le robot est bien à 20cm de l'obstacle.

La fonction du transfert est :

$$V_S = \frac{R_{12}}{R_{11}} * (U_1 - U_2)$$

- **Amplificateur non inverseur :**

On règle le potentiomètre pour limiter la tension à 5V.

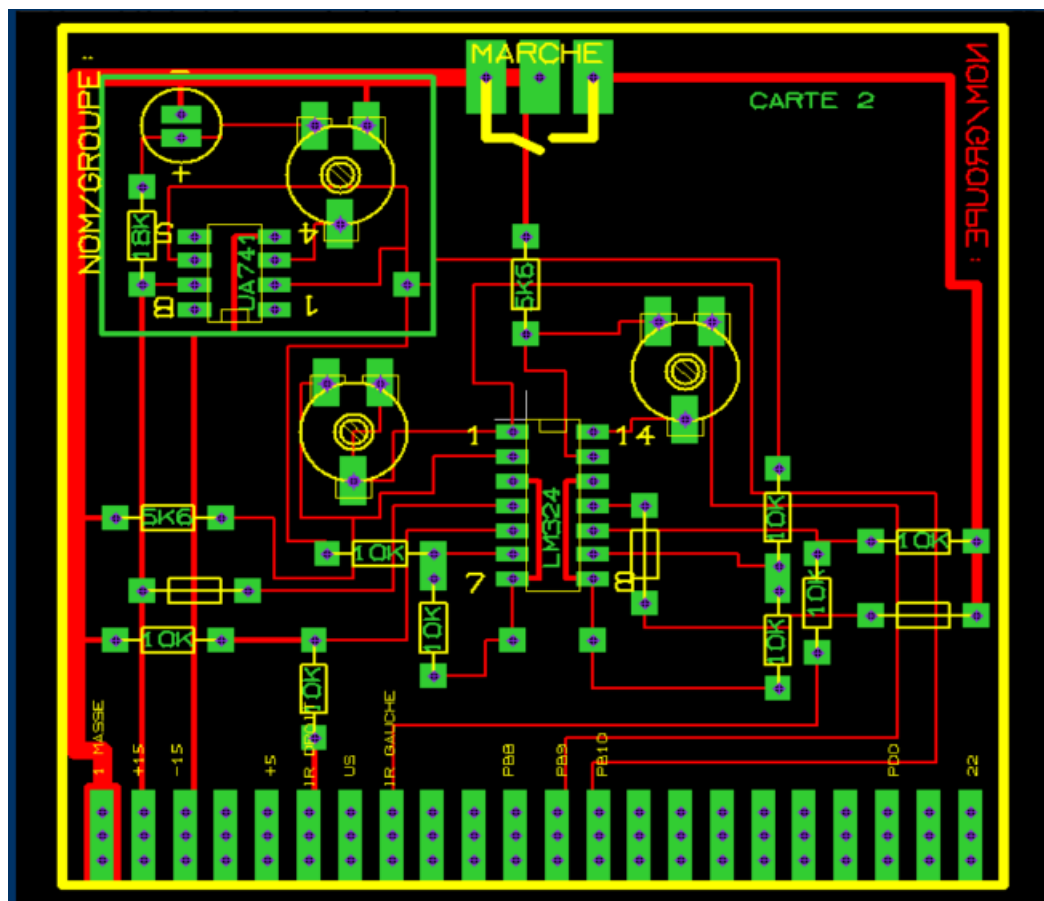
Il est utilisé aussi pour amplifier le signal pour une meilleur vision.

Sa fonction de transfert :

$$V_S = \frac{R_{15} + R_{16}}{R_{15}} * V_e$$

TYPON DE LA CARTE 2

Vous trouvez ci-dessous le typon de la carte 2



PROGRAMATION

- Configuration des entrées sorties :

```
void IO_Init(void)
{
    //Configuration des Entrées/Sorties
    ADPCFG=logique;    //Port B logique-attention ICD3 utilise RB0/RB1 pour PGD/PGC
    TRISA=0x0000;      //Configuration du port A-1:Input
    TRISB=0xFFFF;      //Configuration du port B-0:Output
    TRISC=0xFFFF;      //Configuration du port C-1:Input
    TRISD=0x0001;      //Configuration du port D-1:Input
    TRISE=0x0000;      //Configuration du port E-0:Output
    TRISF=0xFFFF;      //Configuration du port F-1:Input
    TRISG=0xFFFF;      //Configuration du port G-1:Input
}
```

- Déclaration des variables :

```
#include "p30f6010a.h"
#define logique 0xF9EB
#define FCY 5000000
#define PWM 10000

//Codes pour afficheurs 7 segments
#define off 0x00
#define on 0xFF
float mesure,q=5.0/1024 ;
```

- Fonctions utilisées :

```
void avancer(void)
{
    PORTD=0xFF30;    //appel au port D pour les deux roues dans le meme sens
    PDC1=0.8*499;    //vitesse moteur droit
    PDC2=0.8*499;    //vitesse moteur gauche
}
```

```
void stop(void)
{
    PDC1=0;          //ou PDC1=0.5*PTPER; arreter le moteur droit
    PDC2=0;          //ou PDC2=0.5*PTPER; arreter le moteur gauche
}
```

```
void decalage(void)
{
    PDC1=0.8*499+(100*(mesure-0.6));    //ou PDC2=0.5*PTPER; vitesse moteur droit
    PDC2=0.8*499-(100*(mesure-0.6));    //ou PDC2=0.5*PTPER; vitesse moteur gauche
}
```

```

void demi_tour(void)
{
    PORTD=0xFF20;           //appel port d pour les deux roues dans un autre sens

    PDC1=0.8*499;           //ou PDC2=0.5*PTPER; Modification du rapport cyclique moteur droit

    PDC2=0.8*499;           //ou PDC2=0.5*PTPER; Modification du rapport cyclique moteur gauche
}

```

- Programme principal :

```

void main(void)
{
    //Configurations
    IO_Init();               //Appel de la fonction IO_Init() - Entrées-Sorties
    Timer1_Init();           // Configuration du Timer 1
    ADC_Init();              // Configuration ADC
    PWM_Init();              // Configuration Interface PWM
    while(_RD0=1)
    {
        do                  //capteur droit
        {
            mesure=Read_Analog(10)*q; //calcul de la distance entre le mur et le robot droit
            PORTD=0x30;           //sens des roues dans le meme sens
            decalage();           //fonction de decalage
        }
        while(_RB8==0);
        if(_RB8!=0)             //capteur ultrasons
        {
            demi_tour();
            tempo_ms(400);
        }
        else
        {
            PORTD=0xFF30;        //appel port d pour les deux roues dans le meme sens
            avancer();
        }
    }
}

```

Annexe :

- <https://fr.wikipedia.org/>
- www.eureka.univ-lehavre.fr

+ L'aide des professeurs et des collègues du Geii